

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer implemented system for interacting with an object, the system comprising:

at least one processor that executes the following computer executable components stored on at least one computer readable medium:

a method call interceptor that intercepts a method call to an object and that directly routes the method call to an application code generic proxy that is an image of a local object, the method call interceptor accessible to application code to at least one of adapt or extend functionalities of the system; and

~~an~~ the application code generic proxy that receives an intercepted method call, invokes a method on the object without crossing a remoting boundary, receives results from the object and passes results to an entity that generated the intercepted method call based at least in part on the intercepted method call operability of the application code generic proxy modified by the application code, the application code generic proxy performs proxy pre-processing that includes custom, user-directed application processing to at least one of monitor or control the processing of a message between the application code generic proxy and the object that~~machine learning to optimize remote method call invocation before invoking the method on the object, wherein the optimization of the remote method call includes controlling achieving an optimal response time by at least one of determining an optimal amount of data sent to the object, determining an optimal~~ a type of data sent to the object or ~~determining~~ which objects are invoked, in a manner to reduce response time; ~~and~~

a memory operatively coupled to a processor that retains the method call interceptor.

2. (Previously Presented) The system of claim 1, the object is located across a remoting boundary.

3. (Previously Presented) The system of claim 2, the object is marshaled by reference.
4. (Previously Presented) The system of claim 2, the object is marshaled by value.
5. (Previously Presented) The system of claim 1, the method call interceptor populates a call information data store with information associated with the intercepted method call, the call information data store is accessible to the application code generic proxy.
6. (Currently Amended) The system of claim 5, the call information data store is populated with at least one of a method name, one or more input parameters, a count of the number of input parameters, one or more type identifiers associated with the input parameters, a count of the number of return parameters for the method call, one or more type identifiers associated with the return parameters, class[[/]] or interface defining method data, a stack pointer ~~and~~ or a heap pointer.
7. (Previously Presented) The system of claim 6, the call information data store is a message object that can be serialized and passed across a remoting boundary.
8. (Previously Presented) The system of claim 1, the method call interceptor transfers control to a method in the application code generic proxy, the method in the application code generic proxy overrides a base class method defined in a base class object from which the application code generic proxy inherits.
9. (Cancelled).
10. (Currently Amended) The system of claim 1, the proxy pre-processing further comprises at least one of load-balancing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, ~~and~~ or controlling remote method call invocations.
11. (Previously Presented) The system of claim 1, the application code generic proxy

performs proxy post-processing after receiving the results from the object.

12. (Currently Amended) The system of claim 11, the proxy post-processing comprises at least one of auditing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations ~~and~~ or machine learning involved in optimizing remote method call invocation.

13. (Previously Presented) The system of claim 1, the application code generic proxy invokes the method on the object by invoking a method available in a remoting infrastructure.

14. (Currently Amended) A computer readable storage medium containing computer executable components for interacting with an object, the components comprising:

a method call intercepting component that intercepts a method call to an object and directly routes the method call to an application code generic proxy that is an image of a local object, the method call intercepting component accessible to application code to at least one of adapt or extend functionalities of the proxy; and

~~an~~ the application code generic proxy component that receives an intercepted method call, invokes a method on the object without traversing a remoting boundary, receives results from the object and passes results to the entity that generated the intercepted method call, the application code generic proxy component functionality modified by the application code based at least in part on the intercepted method call to perform custom user-directed; ~~the application code generic proxy performs proxy~~ pre-processing that controls the processing of a message between the application code generic proxy and the object, which comprises machine learning to optimize remote method call invocation before invoking the method on the object, wherein the optimizing the remote method call includes achieving a reduced an optimal response time by controlling at least one of ~~determining an optimal~~ amount of data sent to the object, ~~determining an optimal~~ a type of data sent to the object or ~~determining~~ which objects are invoked.

15. (Currently Amended) A method for interacting with an object, the method comprising: employing at least one processor to execute computer executable instructions stored on at least one computer readable medium to perform the following acts:

creating a base class proxy object;
 creating an application code generic proxy, the application code generic proxy inherits from the base class proxy object;
 overriding a base class method defined in the base class, the overridden method receives an intercepted method call;
 intercepting a method call on the object, the interception made accessible to a developer to at least one of adapt or extend functionalities of the application code generic proxy;
 routing the method call to the application code generic proxy without traversing a remoting boundary, wherein the application code generic proxy is an image of a local object;
 adapting the application code generic proxy functionality based at least in part on the method call, the application code generic proxy performs custom user-directed proxy pre-processing comprising transaction processing and machine learning to control ~~optimize remote method call invocation before invoking a method on the object, wherein the optimizing the remote method call includes achieving an optimal response time by~~ at least one of determining an ~~optimal~~ amount of data sent to the object, ~~determining an optimal~~ a type of data sent to the object or ~~determining~~ which objects are invoked;
 invoking the method on the object based in part on the pre-processing;
 performing custom user-directed application processing to at least one of monitor or control the processing of a message between the application code generic proxy and the object;
 receiving a first result from the object; and
 returning a second result to the entity that generated the intercepted method call.

16. (Cancelled).

17. (Currently Amended) The method of claim 15, the proxy pre-processing further comprises at least one of load-balancing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, ~~and~~ or controlling remote method call invocations.

18. (Previously Presented) The method of claim 15, the application code generic proxy performs proxy post-processing before returning the result to the entity that generated the intercepted method call.

19. (Currently Amended) The method of claim 18, the proxy post-processing comprises at least one of auditing, transaction processing, object migration, object persisting, monitoring remote method calls, caching local data, caching remote data, controlling remote method call invocations ~~and~~ or machine learning involved in optimizing remote method call invocation.

20. (Previously Presented) The method of claim 15, the object is located across a remoting boundary.

21. (Previously Presented) The method of claim 20, the object is marshaled by reference.

22. (Previously Presented) The method of claim 20, the object is marshaled by value.

23. (Currently Amended) A computer readable storage medium containing computer executable instructions for performing a method for interacting with an object, the method comprising:

employing at least one processor to execute computer executable instructions stored on at least one computer readable medium to perform the following acts:

creating a base class proxy object;

creating an application code generic proxy, the application code generic proxy inherits from the base class proxy object;

overriding a base class method defined in the base class, the overridden method receives an intercepted method call;

intercepting a method call on the object, the interception made accessible to a developer to at least one of adapt or extend functionalities of the application code generic proxy;

based at least in part on the intercepted method call, adjusting the application code generic proxy functionality;

employing the application code generic proxy to perform custom user-directed

proxy pre-processing comprising at least machine learning ~~to optimize for optimization of~~ remote method call invocation prior to invoking a method on a object such that response time is reduced, wherein the optimization of the remote method call includes controlling achieving an optimal response time by at least one of ~~determining an optimal~~ amount of data sent to the object, ~~determining an optimal~~ type of data sent to the object or ~~determining~~ which objects are invoked;

routing the method call to the application code generic proxy that is an image of a local object, without traversing a remoting boundary;

invoking the method on the object based in part on the custom user-directed proxy pre-processing;

receiving a first result from the object; and

returning a second result to the entity that generated the intercepted method call.

24-26. (Cancelled)

27. (Currently Amended) A computer implemented system for interacting with an object, the system comprising:

at least one processor;

at least one computer readable storage medium storing computer executable instructions that when executed by the at least one processor implement components comprising:

means for creating a base class proxy object from application code, the base class proxy object has a method that can be overridden by an inheriting application code generic proxy so that the overridden method can receive an intercepted method call;

means for creating the application code generic proxy, the application code generic proxy inherits from the base class proxy object and the application code generic proxy overrides the base class method that can be overridden;

means for intercepting a method call and for transferring control to the overridden base class method in the application code generic proxy;

means for the interception to be made accessible to application code to at least one of adapt or extend functionalities of the system;

means for accessing method call interception functionality;

means for retrieving information associated with a method call that can be intercepted by the interception functionality;

means for at least one of adapting or extending the functionality of object systems;

means for the application code generic proxy that is an image of a local object, to receive the intercepted method call without traversing a remoting boundary;

means for modifying the application code generic proxy operability based at least in part on the intercepted method call and externally supplied application code;

means for providing the overridden base class method with a call data structure associated with the intercepted method call;

means for the application code generic proxy to invoke the method on the object;

means for the application code generic proxy to perform proxy pre-processing that includes load-balancing, transaction processing and machine learning to reduce response time ~~optimize remote method call invocation~~ prior to utilizing the means for the application code generic proxy to invoke the method on the object, wherein the proxy pre-processing is user-directed and customized for controlling optimization of the remote method call includes achieving an optimal response time by at least one of determining an optimal amount of data sent to the object, ~~determining an optimal~~ type of data sent to the object or ~~determining~~ which objects are invoked;

means for the application code generic proxy to receive a first result from the object; and

means for the application code generic proxy to return a second result to the entity that generated the intercepted method call

~~wherein a memory operatively coupled to a processor retains at least one of the means.~~